THE UNIVERSITY OF SYDNEY

DISCIPLINE OF BUSINESS ANALYTICS

MPHIL THESIS

Robust Group Testing

Author:

Wenjie Sun

Supervisors:

Dmytro Matsypura

Nam Ho-Nguyen

A thesis submitted in fulfillment of the requirements for the degree of

Master of Philosophy

Discipline of Business Analytics, Business School

University of Sydney

2026

Statement of Originality

I hereby declare that the thesis is written by me in its entirety. This thesis has also not been submitted for any degree in any university previously.

I have duly acknowledged the contribution made to the research by others and all the sources of information which have been used in the thesis.

WENJIE SUN

Acknowledgement

First, I would like to express my sincere gratitude toward my supervisors, Associate Professor Dmytro Matsypura and Nam Ho-Nguyen. Dymtro helps me improve the writing of thesis, and Nam helps me refine the problem as well as write more modular code. Your adherence to academic rigor and detail impresses me a lot. This thesis would not have been possible without your guidance and support.

I would also like to acknowledge my colleagues from the Discipline of Business Analytics and Business school. In particular, I would like to thank my panel members, Professor Daniel Oren and Dr. Li Chen, for your efforts in my progress evaluation meeting. I would also like to thank my fellow students including Johnson, Caroline, Widhiyo, Mike and Michael. Special thank goes to Ben Breeds for your help of solving my admin problems.

Finally, I would like to confirm that the research reported in this thesis is supported by the award of Enhanced Business School Research Scholarship and no content produced by generative AI tools has been used in the preparation of this thesis.

Abstract

Group testing enjoys wide application in the medical diagnosis, compressed sensing and telecommunication. It becomes very popular during the COVID pandemic where test item size is too large to carry out individual testing over each item within the short time period. The main purpose of group testing problem is to minimize the test numbers in detecting defective and non-detective items by designing the corresponding group testing plan. Although there is abundant work on detecting items based on the prescribed test groups, to our best knowledge, the optimization approach to joint group design and testing is limited. In addition, there are numerous uncertainties of the unknown state of items. Thus, we propose a two-stage robust optimization approach to quantifying the uncertainties involved in jointly designing groups and testing items. More specifically, in our second stage problem where we conduct individual testing over items that have not been detected, we build a robust integer optimization model based on a finite scenario uncertainty set of unknown test item states. Furthermore, we develop two methods for solving this model. The first one is a binary programming approach based on lifting the two-stage model with an epigraphical variable. The second one is the Benders' decomposition method because the linear programming relaxation of the second-stage problem with integer variables is tight. Finally, we numerically compare the performances of these two methods under various parameter regimes and uncertainty sets in terms of their computational efficiency.

Contents

Chapter 1

Introduction

The first chapter of this thesis presents the background and motivation, followed by a review of relevant literature.

1.1 Background and Motivation

Group testing (Dorfman, 1943) is a popular problem in various areas, including medical diagnosis, compressed sensing, and combinatorial designs. It aims to minimize the number of tests in finding all defective items. Du and Hwang (1999) study the screening for a rare disease called syphilis, and they find that it is too time consuming and costly to carry out individual tests when the sample size is large and most of them are negative. However, performing group testing can help us spend less time and money in screening for this disease. If we obtain a negative result from group testing for a subset, it implies that all samples of that set are negative so that there is no need to conduct individual tests. If we obtain a positive test for a subset, we carry out the individual test of that subset to find the defective items. Group testing also enjoys many applications beyond medical diagnosis. For example, Malioutov and Malyutov (2012) conduct a small number of boolean group testing in compressed sensing to discover a sparse high-dimensional boolean vector with Bernoulli noise. Colbourn and Dinitz (2006) form a group with the Bernoulli design based on random samples from the Bernoulli distribution. Although group testing can save a lot of time and money spent on individual testing, it's impossible for group testing to detect all defective and non-defective items. More specifically, some defective items cannot be detected by group testing if there are other defective items in the same group. Some non-defective items cannot be detected by group testing if there are other defective items in the same group. Thus, in order to detect all items, we need to conduct individual testing on the items that cannot be detected by the group testing.

In practice, the states of the items in the testing groups are often unknown to us. Thus, group testing problem can be characterized by the optimization model under uncertainty. More specifically, when the states of defective items follow some prescribed distribution, we can model group testing problem as a stochastic optimization problem which is to minimize the expected number of tests. When the states of defective items belong to an ambiguity set, we can model group testing problem as a robust optimization problem which is to minimize the worst-case expected number of tests over the constructed ambiguity set. Furthermore, because we may not be able to detect all items in the group testing, we need to formulate a two-stage stochastic or robust optimization model to detect all items using both individual testing and group testing while minimizing the total test number. For the two-stage stochastic optimization problem, a commonly used solution approach is called L-shaped method. The main principle of this method is to generate a sequence of cutting planes called Bender cuts to approximate the recourse cost function. This method is also called outer linearization or the Benders' decomposition method (Benders, 1962). A comprehensive review of the Benders' decomposition method can be found in Rahmaniani et al. (2017). Correspondingly, Dantzig-Wolf decomposition (Dantzig and Wolfe, 1960) also known as the inner linearization is another kind of cutting plane method for solving the two-stage stochastic optimization problem.

In this thesis, considering that it's difficult to characterize the distribution of the states of unknown items, we propose a two-stage robust optimization problem for joint group design and testing. In the first stage problem, we design the test set for each item and do group testing over all items. In the second stage problem, given a discrete uncertainty set for the unknown states of test samples, we do individual testing over the items uncertified by group testing stage. Then, we can reformulate our two-stage problem as a binary programming problem by adding an epigraphical variable. Alternatively, because the second stage problem with integer variables is equivalent to its linear programming relaxation counterpart, the two-stage problem can be tractably solved by the Benders' decomposition method. Finally, we conduct several numerical experiments to compare the performance of these two methods for solving the two–stage robust optimization model under different uncertainty sets.

Our main contributions are summarized as follows.

- We develop a two-stage robust optimization model for joint group design and testing.
- We prove the tightness of the linear programming relaxation of the second-stage problem by exploiting its structure.

- Based on the tightness of linear programming relaxation, we solve our problem using the Benders' decomposition method.
- We conduct numerical experiments to compare the performances of the binary programming and the Benders' decomposition method for solving the two-stage robust optimization model under various parameter regimes and uncertainty sets.

1.2 Literature Review

Group testing problem can be divided into adaptive and non-adaptive ones. Non-adaptive group testing is conducted in two separate stages including the design stage and the test stage (Aldridge et al., 2019; Coja-Oghlan et al., 2021). The design stage determines the testing pools, followed by the test stage in which we certify which items were defective or non-defective. However, adaptive group testing (D'yachkov et al., 2016; Vorobyev, 2019) alternates between the design step and the test step by designing future tests based on previous test outcomes. Furthermore, group testing problem can also be divided into noise and noiseless ones, depending on whether the test results are accurate or not. Unlike the noiseless setting, the noisy testing relaxes the assumption that tests can perfectly identify the presence of defective items in the pool (Scarlett, 2019). There are also many other classifications for group testing problems, e.g., binary vs. non-binary outcomes, known vs. unknown number of defectives, and sparse regime vs. linear regime. In this thesis, we focus on the noiseless non-adaptive group testing problem.

There are different approaches to the non-adaptive group testing problem. Many existing works deal with group testing based on information theory and combinatorics (Barg and Mazumdar, 2015). Aldridge et al. (2019) surveys the algorithmic and information-theoretic approach to the non-adaptive setting with a small (non-zero) error probability in various sparsity regimes. Some group testing algorithms include the combinatorial optimal matching pursuit (COMP) algorithm, the definite defectives algorithm (DD), and the sequential COMP (SCOMP) algorithm. The COMP algorithm is like the non-conservative two-stage testing where we discover both defective and non-defective items in the first stage while the DD algorithm is like the conservative two-stage testing where we only discover non-defective items in the first stage. Some other existing works focus on developing the optimization-based approach to group testing problem. Chan et al. (2012) and Malioutov and Malyutov (2012) use linear programming models to certify defective items based on grouping test results. Compared with the information-theoretic approach which can only obtain the asymptotically optimal test design under particular parameter regimes, the optimization-based approach can obtain the exact optimal design under whichever parameter regimes. Thus, in this thesis, we propose the integer programming approach to certifying defective and non-defective items in the group testing procedure based on the definition of definitely defective and definitely non-defective items (Aldridge et al., 2019). However, most literature on the optimization-based approach to group testing doesn't consider how to design testing groups. To our knowledge, Long et al. (2022) is the only relevant paper that studies the noisy group test problem with the ambiguity of population distribution under full and partial correlation. Their optimal test design minimizes the worst-case errors and weighted counts of tests where there is correlation between individuals and their prevalence rates.

1.3 Structure

In Chapter 2, we propose a two-stage robust optimization model for joint group design and testing based on an integer programming formulation for calculating the total number of certified defective and non-defective items. In Chapter 3, we develop a Benders' decomposition method based on a tight linear programming relaxation of the second stage problem. In Chapter 4, we conduct numerical experiments to compare the performances of the binary programming and the Benders' decomposition method under various parameter regimes and uncertainty sets.

Chapter 2

Two-stage Robust Optimization Model

In this chapter, we propose a two-stage robust optimization for the joint group testing design and a corresponding binary programming solution approach. First, we define the certified defective and non-defective items. Then, based on the definitions, we propose an integer programming model to calculate the total number of certified defective and non-defective items in group testing problem. Because group testing may not be able to detect all items, we formulate a two-stage robust optimization model consisting of both individual testing and group testing stages to detect all defective and non-defective items. Finally, we propose a binary programming solution approach to solve that model under a discrete uncertainty set.

2.1 Problem Definition and Notation

We consider the following two-stage group testing of n items with defective states $\{d_i\}_{i\in[n]} \subseteq \{0,1\}^n$. In the first stage, for each test $t \in [T]$, the group $S_t \subseteq [n]$ being formed and tested leads to the observed result $y_t \in \{0,1\}$. In the second stage, we perform individual testing of uncertified items. The true states of items are denoted by

$$d_i = \begin{cases} 0, & \text{if item } i \text{ is non-defective,} \\ 1, & \text{if item } i \text{ is defective,} \end{cases} \qquad i \in [n].$$

The state of each item belongs to a subset D of $\{0,1\}^n$. The matrix $X \in \{0,1\}^{T\times n}$ describes which elements are part of which tests as follows.

$$x_{ti} = \begin{cases} 0, & \text{item } i \text{ is not part of test } t, \\ 1, & \text{item } i \text{ is part of test } t, \end{cases} \quad t \in [T], i \in [n].$$

Given test design X and state d, group testing output is $y(X,d) = \{y_t(X,d)\}_{t\in[T]} \subseteq \{0,1\}^T$ where $y_t(X,d) = \max_{i\in[n]} d_i x_{ti}$ for each $t \in [T]$. In the group testing literature (Aldridge et al., 2019), certified items include both the definitely non-defective items that are in a test group only consisting of non-defective items and the definitely defective items that are in a test group where all other items have been certified as non-defective. This definition is useful for detecting defective items because in a positive test with more than one items that haven't been certified as definitely non-defective, we can't determine whether these items are defective or non-defective. Then, we mathematically define what we mean by certifying that an item is defective or non-defective.

Definition 1. Given the fixed number T of group tests, any item that appears in a negative tests, e.g., $y_t(X,d) = 0$ for some $t \in [T]$, is called a certified non-defective (CND), and any item that appears in a positive test, e.g., $y_t(X,d) = 1$ for some $t \in [T]$, where every other item is called a certified defective (CD).

Given group tests X and outcomes y, let $H_{\text{CD}}(X,y)$ be the number of certified defective items and $H_{\text{CND}}(X,y)$ the number of certified non-defective items. Since outcomes y depend on both X and d, we define the total number of certified items as

$$H(X,y) \triangleq H_{\text{CD}}(X,y) + H_{\text{CND}}(X,y),$$

the negative test set as

$$H_{T_0}(y) = \{t \in [T] | y_t = 0\},$$

and the positive test set as

$$H_{T_1}(y) = \{t \in [T] | y_t = 1\}.$$

In the second stage, we can only certify the non-defective elements of the group $t \in H_{T_0}(y)$ and defective components that do not appear with other certified non-defective elements. Then, we propose the following mathematical definition of certified defective and non-defective items:

Definition 2. The set of certified non-defective items is

$$\text{CND} = \{i \in [n] | \exists t \in H_{T_0}(y) \quad \text{s.t.} \quad x_{ti} = 1\}, \tag{2.1}$$

and the set of certified defective items is

$$\text{CD} = \{i \in [n] | \exists t \in H_{T_1}(y) \quad \text{s.t.} \quad x_{ti} = 1, x_{tj} = 0, \forall j \in [n] \setminus (\{i\} \cup \text{CND})\}. \tag{2.2}$$

2.2 An Integer Programming Model for Certifying Items

We then characterize the number of certified defective items $H_{\mathrm{CND}}(y)$ and the number of certified non-defective items $H_{\mathrm{CD}}(y)$ using the following integer programming models:

$$\max_{z_i^0, z_{ti}^0} \quad \sum_{i \in [n]} z_i^0 \tag{2.3a}$$

$$\text{s.t.} \quad z_{ti}^0 \leq 1 - y_t, \forall t \in [T], i \in [n], \tag{2.3b}$$

$$z_{ti}^0 \leq x_{ti}, \forall t \in [T], i \in [n], \tag{2.3c}$$

$$z_i^0 \leq \sum_{t \in [T]} z_{ti}^0, \forall i \in [n], \tag{2.3d}$$

$$z_{ti}^0 \in \{0, 1\}, \forall t \in [T], i \in [n], \tag{2.3e}$$

$$z_i^0 \in \{0, 1\}, \forall i \in [n]. \tag{2.3f}$$

where $\{z_i^0\}_{i \in [n]}$ aim to certify the non-defectiveness of item i, $\{z_{ti}^0\}_{t \in [T], i \in [n]}$ aim to certify the existence of item i in a negative test t, the objective function (2.3a) denotes the number of certified non-defective items, and constraints (2.3b) and (2.3c) denote whether item i could belong to a negative testing group. More specifically, item i is in a negative testing group t if and only if $y_t = 0$ and $x_{ti} = 1$. Thus, constraints (2.3d) mean that item i is a certified non-defective item when $z_i^0 = 1$ at the optimality if and only if it is in at least one of negative testing groups for which $z_{ti}^0 = 1$ at the optimality.

Given the optimal solutions $\{z_i^{0*}\}_{i \in [n]}$ to Problem (2.3),

$$\max_{z_i^1, z_{ti}^1} \quad \sum_{i \in [n]} z_i^1 \tag{2.4a}$$

$$\text{s.t.} \quad z_{ti}^1 \leq x_{ti}, \forall t \in [T], i \in [n], \tag{2.4b}$$

$$z_{ti}^1 \leq \sum_{j \in [n]} d_j x_{tj}, \forall t \in [T], i \in [n], \tag{2.4c}$$

$$z_{ti}^1 \leq 1 - x_{tj} + z_j^{0*}, \forall t \in [T], i \in [n], j \in [n] \setminus \{i\}, \tag{2.4d}$$

$$z_i^1 \leq \sum_{t \in [T]} z_{ti}^1, \forall i \in [n], \tag{2.4e}$$

$$z_{ti}^1 \in \{0, 1\}, \forall t \in [T], i \in [n], \tag{2.4f}$$

$$z_i^1 \in \{0, 1\}, \forall i \in [n]. \tag{2.4g}$$

where $\{z_i^1\}_{i \in [n]}$ aim to certify the defectiveness of item i, $\{z_{ti}^1\}_{t \in [T], i \in [n]}$ aim to certify the existence of item i in a positive test t, the objective function (2.4a) denotes the number of certified defective items, and constraints (2.4b-2.4d) denote whether item i could belong to a positive testing group. More specifically, item i is in a positive testing group t if and only if

$y_t = 1$ or $\sum_{j\in[n]} d_j x_{tj} = 1$ and $x_{ti} = 1$. Constraints (2.4d) denote whether item i is in a testing group t without other defective or uncertified non-defective items. Thus, constraints (2.4e) mean that item i is a certified defective item when $z_i^1 = 1$ at the optimality if and only if it is in at least one of these positive testing groups for which $z_{ti}^1 = 1$ at the optimality. Then, we formally prove the above arguments in the following proposition:

Proposition 1. $H_{\text{CD}}(X, y)$ and $H_{\text{CND}}(X, y)$ are, respectively, equal to the optimal values of Problems (2.3) and (2.4).

Proof. We prove both (2.3) and (2.4).

1. We first want to show that $H_{\text{CND}}(X, y) = |\text{CND}|$ can be derived from Problem (2.3) based on the following cases:

   (a) $x_{ti} = 1$ and $y_t = 1 \Rightarrow z_{ti}^0 = 0$.

   (b) $x_{ti} = 0$ and $y_t = 1 \Rightarrow z_{ti}^0 = 0$.

   (c) $x_{ti} = 1$ and $y_t = 0 \Rightarrow z_{ti}^0$ can be either 0 or 1.

   (d) $x_{ti} = 0$ and $y_t = 1 \Rightarrow z_{ti}^0 = 0$.

   After enumerating all cases, we have that at optimality

   (a) $x_{ti} = 1$ and $y_t = 1 \Rightarrow z_{ti}^0 = 0$.

   (b) $x_{ti} = 0$ and $y_t = 1 \Rightarrow z_{ti}^0 = 0$.

   (c) $x_{ti} = 1$ and $y_t = 0 \Rightarrow z_{ti}^0 = 1$.

   (d) $x_{ti} = 0$ and $y_t = 0 \Rightarrow z_{ti}^0 = 0$.

   Then, we see that there exists $t \in [T]$ such that at optimality $z_{ti}^0 = 1$ if and only if $i \in \text{CND}$. Furthermore, we have that, at optimality $z_i^0 = 1$ if and only if there exists $t \in [T]$ such that $z_{ti}^0 = 1$. Thus, we have that, at optimality $H_{\text{CND}} = \sum_{i\in[n]} z_i^{0*}$ where $\{z_i^{0*}\}_{i\in[n]}$ are the optimal solutions to Problem (2.3).

2. Next we want to show that at optimality $z_i^1 = 1$ if and only if $i \in \text{CD}$. By the following constraints:

   $$z_{ti}^1 \le x_{ti}, z_{ti}^1 \le \sum_{j\in[n]} d_j x_{tj}, z_{ti}^1 \le 1 - x_{tj} + z_j^{0*}, \forall t \in [T], j \in [n] \setminus \{i\}, i \in [n], z_i^1 \in \{0, 1\},$$

   we have that at optimality $z_{ti}^1 = 1$ if and only if the following three conditions hold simultaneously:

- $\sum_{j\in[n]} d_j x_{tj} \geq 1$,
- $x_{ti} = 1$,
- $\sum_{j\in[n]\setminus\{i\}} x_{tj}(1 - z_j^{0*}) = 0$.

Then, since, by the definition of certified items,

- $y_t = 1 \Leftrightarrow \sum_{j\in[n]} d_j x_{tj} \geq 1$,
- $x_{ti} = 1$,
- $x_{tj} = 0$ or $z_j^{0*} = 1$, $\forall j \in [n] \setminus \{i\} \Leftrightarrow \sum_{j\in[n]\setminus\{i\}} x_{tj}(1 - z_j^{0*}) = 0$,

we have that at optimality $z_{ti}^1 = 1$ if and only if $i \in \mathrm{CD}$. Finally, because $z_i^1 \leq \sum_{t\in[T]} z_{ti}^1$ and $z_i^1 \in \{0, 1\}$ for each $i \in [n]$, we have that at optimality $z_i^1 = 1$ if and only if $i \in \mathrm{CD}$. Thus, we conclude that at optimality $H_{\mathrm{CD}} = \sum_{i\in[n]} z_i^{1*}$ where $\{z_i^{1*}\}_{i\in[n]}$ are the optimal solutions to Problem (2.4).

□

Proposition 1 connects Definitions 2.1 and 2.2 with the objective functions in Problems (2.3) and (2.4). Intuitively, constraint (2.3b) means that it's possible that item i can be in a negative test t, constraint (2.3c) denotes whether item i belongs to test t, and constraint (2.3d) denotes whether item i can be in a negative test t. Thus, it implies that the objective function (2.3a) can be the total number of non-defective items in the group test t. Similar reasoning process applies to the connection between Definition 2.2 and Problem (2.4). However, it's impossible for group testing to detect all defective and non-defective items because some defective items could be in the same positive testing group and some non-defective items could be in positive testing groups. Thus, it's necessary to conduct individual testing over all the items that can't be certified by the group testing.

2.3 Two-stage Robust Optimization Model for Designing Optimal Testing Group

In order to minimize the total number of group and individual tests, we formulate the following optimization model:

$$
\begin{aligned}
\min_{X,b} \quad & \sum_{t\in[T]} b_t + n - H(X,y) && \text{(2.5a)}\\
\text{s.t.} \quad & \sum_{i\in[n]} x_{ti} \le Ub_t, \forall t \in [T], && \text{(2.5b)}\\
& \sum_{t\in[T]} x_{ti} \le r, \forall i \in [n], && \text{(2.5c)}\\
& x_{ti} \in \{0,1\}, \forall t \in [T], i \in [n], && \text{(2.5d)}\\
& b_t \in \{0,1\}, \forall t \in [T], && \text{(2.5e)}
\end{aligned}
$$

where $b_t \in \{0,1\}$ denotes whether the group test t should be conducted, U is the upper bound for the size of the group and r is the upper bound for the number of each item tested. In practice such as COVID pandemic, it's difficult to get the prior knowledge of the real states of test sample like patients' real state. Here, given the unknown real states $\{d_i\}_{i\in[n]}$, we consider the following two-stage robust optimization problem under the uncertainty set D:

$$
\begin{aligned}
\min_{X,b} \quad & \sum_{t\in[T]} b_t + n - \min_{d\in D} Q(X,d) && \text{(2.6a)}\\
\text{s.t.} \quad & \sum_{i\in[n]} x_{ti} \le Ub_t, \forall t \in [T], && \text{(2.6b)}\\
& \sum_{t\in[T]} x_{ti} \le r, \forall i \in [n], && \text{(2.6c)}\\
& x_{ti} \in \{0,1\}, \forall t \in [T], i \in [n], && \text{(2.6d)}\\
& b_t \in \{0,1\}, \forall t \in [T], && \text{(2.6e)}
\end{aligned}
$$

where $Q(X,d)$ denotes the number of items certified by group testing and is equivalent to $H(X,y)$ because the outcome y is a function of d. Uncertainty set D can take various forms, e.g., $D = \{d \in \{0,1\}^n | \sum_{i=1}^n d_i = m\}$ and $D = \{d \in \{0,1\}^n | d \overset{i.i.d.}{\sim} \text{Bernoulli}(p)\}$ where p is the probability of item i being defective, depending on the prior information we have known about the true defective state d. Then, we propose the following integer programming model to calculate $Q(X,d)$.

$$\max_{z_i^0, z_i^1, z_{ti}^0, z_{ti}^1} \quad \sum_{i\in[n]} z_i^0 + z_i^1 \tag{2.7a}$$

$$\text{s.t.} \quad z_{ti}^0 \le x_{ti}, \forall t \in [T], i \in [n], \tag{2.7b}$$

$$z_{ti}^0 \le 1 - d_j x_{tj}, \forall t \in [T], i \in [n], j \in [n], \tag{2.7c}$$

$$z_i^0 \le \sum_{t\in[T]} z_{ti}^0, \forall i \in [n], \tag{2.7d}$$

$$z_{ti}^1 \le x_{ti}, \forall t \in [T], i \in [n], \tag{2.7e}$$

$$z_{ti}^1 \le \sum_{j\in[n]} d_j x_{tj}, \forall t \in [T], i \in [n], \tag{2.7f}$$

$$z_{ti}^1 \le 1 - x_{tj} + z_j^0, \forall t \in [T], j \in [n] \setminus \{i\}, i \in [n], \tag{2.7g}$$

$$z_i^1 \le \sum_{t\in[T]} z_{ti}^1, \forall i \in [n], \tag{2.7h}$$

$$z_{ti}^0 \in \{0,1\}, \forall t \in [T], i \in [n], \tag{2.7i}$$

$$z_{ti}^1 \in \{0,1\}, \forall t \in [T], i \in [n], \tag{2.7j}$$

$$z_i^0 \in \{0,1\}, \forall i \in [n], \tag{2.7k}$$

$$z_i^1 \in \{0,1\}, \forall i \in [n]. \tag{2.7l}$$

Based on Proposition 1, we have the following proposition:

Proposition 2. $Q(X,d)$ is equal to the optimal value of Problem (2.7).

Proof. On one hand, Problem (2.7) can be sequentially decomposed into Problems (2.4) and (2.3) whose optimal values are respectively equal to H_{CD} and $H_{\text{CND}}(X,y)$. On the other hand, $Q(X,d) \triangleq H(X,y)$ and $H(X,y) \triangleq H_{\text{CD}}(X,y) + H_{\text{CND}}(X,y)$. Thus, we get the conclusion. □

Proposition 2 shows our optimization approach to group testing can calculate all number of certified defective and non-defective items. When $D = \{d^1, \ldots, d^K\}$, by incorporating the auxiliary decision variable R, we reformulate Problem (2.5) as the following binary programming problem:

$$
\begin{aligned}
\min_{X,b,R} \quad & \sum_{t\in[T]} b_t + n - R && \text{(2.8a)}\\
\text{s.t.} \quad & \sum_{i\in[n]} x_{ti} \le U b_t, \forall t \in [T], && \text{(2.8b)}\\
& \sum_{t\in[T]} x_{ti} \le r, \forall i \in [n], && \text{(2.8c)}\\
& Q(X,d) \ge R, \forall d \in D, && \text{(2.8d)}\\
& x_{ti} \in \{0,1\}, \forall t \in [T], i \in [n], && \text{(2.8e)}\\
& b_t \in \{0,1\}, \forall t \in [T]. && \text{(2.8f)}
\end{aligned}
$$

Here, the key issue is how to deal with the constraint (2.8d) involving the complicated $Q(X,d)$.

2.4 Binary Programming Approach

We reformulate the constraints $Q(X,d) \ge R$ for all $d \in D$ as the explicit form based on Problem (2.7). An approach is to find some feasible solutions to Problem (2.7) such that its objective function value is larger than or equal to R. This approach works well when K is small due to the resulting small number of binary variables and constraints. More specifically, Problem (2.8) can be reformulated as the following binary programming model:

$$\min_{x_{ti},b_t,R,z_{ti}^{0,k},z_{ti}^{1,k},z_i^{0,k},z_i^{1,k}} \quad \sum_{t\in[T]} b_t + n - R \tag{2.9a}$$

$$\text{s.t.} \quad \sum_{i\in[n]} x_{ti} \le Ub_t, \forall t \in [T], \tag{2.9b}$$

$$\sum_{t\in[T]} x_{ti} \le r, \forall i \in [n], \tag{2.9c}$$

$$\sum_{i\in[n]} z_i^{0,k} + z_i^{1,k} \ge R, \forall k \in [K], \tag{2.9d}$$

$$z_{ti}^{0,k} \le x_{ti}, \forall t \in [T], i \in [n], k \in [K], \tag{2.9e}$$

$$z_{ti}^{0,k} \le 1 - d_j^k x_{tj}, \forall t \in [T], i \in [n], k \in [K], j \in [n], \tag{2.9f}$$

$$z_i^{0,k} \le \sum_{t\in[T]} z_{ti}^{0,k}, \forall i \in [n], k \in [K], \tag{2.9g}$$

$$z_{ti}^{1,k} \le x_{ti}, \forall t \in [T], i \in [n], k \in [K], \tag{2.9h}$$

$$z_{ti}^{1,k} \le \sum_{j\in[n]} d_j^k x_{tj}, \forall t \in [T], i \in [n], k \in [K], \tag{2.9i}$$

$$z_{ti}^{1,k} \le 1 - x_{tj} + z_j^{0,k}, \forall t \in [T], j \in [n] \setminus \{i\}, i \in [n], k \in [K], \tag{2.9j}$$

$$z_i^{1,k} \le \sum_{t\in[T]} z_{ti}^{1,k}, \forall i \in [n], k \in [K], \tag{2.9k}$$

$$z_{ti}^{0,k} \in \{0,1\}, \forall t \in [T], i \in [n], k \in [K], \tag{2.9l}$$

$$z_{ti}^{1,k} \in \{0,1\}, \forall t \in [T], i \in [n], k \in [K], \tag{2.9m}$$

$$z_i^{0,k} \in \{0,1\}, \forall i \in [n], k \in [K], \tag{2.9n}$$

$$z_i^{1,k} \in \{0,1\}, \forall i \in [n], k \in [K], \tag{2.9o}$$

$$x_{ti} \in \{0,1\}, \forall t \in [T], i \in [n], \tag{2.9p}$$

$$b_t \in \{0,1\}, \forall t \in [T]. \tag{2.9q}$$

However, in Problem (2.9), the computational difficulty will be exacerbated by the large number of binary variables and constraints especially when K, T and n become larger.

Chapter 3

Benders' Decomposition Method

In this chapter, we aim to develop efficient algorithm to solving Problem (2.6). We first show the tightness of linear programming relaxation of Problem (2.7) and equivalently reformulate it as a single stage problem with Bender cuts. Then, we propose our Benders' decomposition method for solving Problem (2.6).

3.1 Linear Programming Relaxation of Problem (2.7)

In order to get around the computational difficulty in Problem (2.9), we consider the following linear programming relaxation of Problem (2.7):

$$\max_{z_i^0, z_i^1, z_{ti}^0, z_{ti}^1} \quad \sum_{i \in [n]} z_i^0 + z_i^1 \tag{3.1a}$$

$$\text{s.t.} \quad z_{ti}^0 \leq x_{ti}, \forall t \in [T], i \in [n], \tag{3.1b}$$

$$z_{ti}^0 \leq 1 - d_j x_{tj}, \forall t \in [T], i \in [n], j \in [n], \tag{3.1c}$$

$$z_i^0 \leq \sum_{t \in [T]} z_{ti}^0, \forall i \in [n], \tag{3.1d}$$

$$z_{ti}^1 \leq x_{ti}, \forall t \in [T], i \in [n], \tag{3.1e}$$

$$z_{ti}^1 \leq \sum_{j \in [n]} d_j x_{tj}, \forall t \in [T], i \in [n], \tag{3.1f}$$

$$z_{ti}^1 \leq 1 - x_{tj} + z_j^0, \forall t \in [T], j \in [n] \setminus \{i\}, i \in [n], \tag{3.1g}$$

$$z_i^1 \leq \sum_{t \in [T]} z_{ti}^1, \forall i \in [n], \tag{3.1h}$$

$$z_{ti}^0 \leq 1, \forall t \in [T], i \in [n], \tag{3.1i}$$

$$z_{ti}^1 \leq 1, \forall t \in [T], i \in [n], \tag{3.1j}$$

$$z_i^0 \leq 1, \forall i \in [n], \tag{3.1k}$$

$$z_i^1 \leq 1, \forall i \in [n], \tag{3.1l}$$

$$z_{ti}^0 \geq 0, \forall t \in [T], i \in [n], \tag{3.1m}$$

$$z_{ti}^1 \geq 0, \forall t \in [T], i \in [n], \tag{3.1n}$$

$$z_i^0 \geq 0, \forall i \in [n], \tag{3.1o}$$

$$z_i^1 \geq 0, \forall i \in [n]. \tag{3.1p}$$

Because $z_i^0 = z_i^1 = 0$ for $i \in [n]$ are always feasible solutions to Problem (3.1) independent from the state $d \in \{0,1\}^n$ and test design $X \in \{0,1\}^{T \times n}$. In addition, the following proposition shows that Problem (3.1) is a tight linear programming relaxation of Problem (2.7).

Proposition 3. The optimal value of Problem (3.1) is equal to $Q(X, d)$.

Proof. We want to show that the optimal value of Problem (3.1) is the total number of certified and non-certified items. More specifically, we want to show that at optimality $z_i^{0*} = 1$ when $i \in \text{CND}$ and $z_i^{1*} = 1$ when $i \in \text{CD}$.

- $i \in \text{CND}$: There must exist $t \in [T]$ such that $y_t = 0$ and $x_{ti} = 1$. It follows that $d_i x_{ti} = 0$

for all $i \in [n]$. Then, by (3.1b) and (3.1c), at optimality we can let $z_{ti}^{0*} = 1$. Thus, by (3.1d), we conclude that at optimality $z_i^{0*} = 1$.

- $i \notin \text{CND}$: For all $t \in [T]$ such that $y_t = 0$, we have $x_{ti} = 0$. Then, by (3.1b), for all $t \in [T]$ such that $y_t = 0$, we have $z_{ti}^0 = 0$. For all $t \in [T]$ such that $y_t = 0$, there must exist $i \in [n]$ such that $d_i x_{ti} = 1$. Then, by (3.1c), for all $t \in [T]$ such that $y_t = 0$, we have $z_{ti}^0 = 0$. Thus, by (3.1d), we conclude that at optimality $z_i^{0*} = 0$.

- $i \in \text{CD}$: There must exist $t \in [T]$ such that $y_t = 1$, $x_{ti} = 1$ and $x_{tj} = 0$ for all $j \in [n] \setminus (\{i\} \cup \text{CND})$. Then, by the definition of certified items, we have $x_{ti} = 1$, $d_i = 1$, and $\sum_{j \in [n] \setminus \{i\}} x_{tj}(1 - z_j^{0*}) = 0$. By (3.1e), (3.1f) and (3.1g), we have at optimality $z_{ti}^{0*} = 1$ for all $t \in [T]$. Thus, by (3.1d), we can conclude that at optimality $z_i^{0*} = 1$.

- $i \notin \text{CD}$: For all $t \in [T]$ such that $y_t = 1$, we have $x_{ti} = 0$ or $x_{tj} = 1$ for some $j \in [n] \setminus (\{i\} \cup \text{CND})$. When $x_{ti} = 0$, by (3.1e), we have $z_{ti}^1 = 0$. When $x_{tj} = 1$ for some $j \in [n] \setminus (\{i\} \cup \text{CND})$, by (3.1g), we have $z_{ti}^1 = 0$ because $x_{tj} = 1$ and $z_j^{0*} = 0$. Then, based on above two cases, for all $t \in [T]$ such that $y_t = 1$, we have $z_{ti}^{1*} = 0$. For all $t \in [T]$ such that $y_t = 0$, we have $\sum_{j \in [n]} d_j x_{tj} = 0$, and thus by (3.1f), $z_{ti}^{1*} = 0$. Thus, by (3.1h), we can conclude that $z_i^{1*} = 0$.

In summary, because by Propositions 1 and 2, the total number of certified and non-certified items is equal to $Q(X, d)$, we get the conclusion. $\square$

Proposition 3 is intuitively right because in Problem (2.7), at the optimality, both the left hand side and the right hand side of constraints (2.7b) – (2.7h) should take as large value as possible within the interval $[0, 1]$. In addition, Proposition 3 implies that Problem (2.7) can be solved by the simplex algorithm within the polynomial time. Hence, since Problem (2.7) can be equivalently reduced to a linear programming problem, Problem (2.6) can be tractably solved by the Benders' decomposition method.

3.2 Benders' Master Problem

In order to derive the master problem, we first assign dual variables to constraints in Problem (3.1) as follows.

Primal Constraint	Dual Variable	Sign
$z^0_{ti} \leq x_{ti}$	u^0_{ti}	≥ 0
$z^0_{ti} \leq 1 - d_j x_{tj}$	u^0_{tij}	≥ 0
$z^0_i \leq \sum_t z^0_{ti}$	ξ_i	≥ 0
$z^1_{ti} \leq x_{ti}$	u^1_{ti}	≥ 0
$z^1_{ti} \leq \sum_j d_j x_{tj}$	u^2_{ti}	≥ 0
$z^1_{ti} \leq 1 - x_{tj} + z^0_j$	u^1_{tij}	≥ 0
$z^1_i \leq \sum_t v_{ti}$	η_i	≥ 0
$z^0_i \leq 1$	μ^0_i	≥ 0
$z^1_i \leq 1$	μ^1_i	≥ 0
$z^0_{ti} \leq 1$	v^0_{ti}	≥ 0
$z^1_{ti} \leq 1$	v^1_{ti}	≥ 0

Then, we multiply the right hand side of each primal constraint in Problem (3.1) by its corresponding dual variable and sum them as follows.

$$\begin{aligned}
&\sum_{t,i}(u^0_{ti} + u^1_{ti})x_{ti} + \sum_{t,i,j} u^0_{tij}(1 - d_j x_{tj}) + \sum_{t,i} u^2_{ti}\left(\sum_j d_j x_{tj}\right) + \sum_{t,i,j\neq i} u^1_{tij}(1 - x_{tj}) \\
&+ \sum_{t,i}(v^0_{ti} + v^1_{ti}) + \sum_i(\mu^0_i + \mu^1_i) \\
&\geq \sum_{t,i}\left(u^0_{ti} + \sum_{j\in[n]} u^0_{tij} + v^0_{ti} - \xi_i\right)z^0_{ti} + \sum_{t,i}\left(u^1_{ti} + u^2_{ti} + \sum_{j\in[n]\setminus\{i\}} u^1_{tij} + v^1_{ti} - \eta_i\right)z^1_{ti} \\
&\quad + \sum_i\left(v^0_i - \sum_{t\in[T],j\in[n]\setminus\{i\}} u^1_{tij} + \xi_i\right)z^0_i + \sum_i(v^1_i + \eta_i)z^1_i.
\end{aligned}$$

In order to form an upper bound on the primal objective in Problem (3.1), we let

$$\begin{aligned}
&u^0_{ti} + \sum_{j\in[n]} u^0_{tij} + v^0_{ti} - \xi_i \geq 0, \forall t \in [T], i \in [n], \\
&u^1_{ti} + u^2_{ti} + \sum_{j\in[n]\setminus\{i\}} u^1_{tij} + v^1_{ti} - \eta_i \geq 0, \forall t \in [T], i \in [n], \\
&\xi_i + v^0_i - \sum_{t,j\neq i} u^1_{tij} \geq 1, \forall i \in [n], \\
&\eta_i + v^1_i \geq 1, \forall i \in [n].
\end{aligned}$$

Thus, after taking the dual of Problem (3.1), we have

$$\min_{v_i^0, v_i^1, \xi_i, \eta_i, u_{ti}^0, u_{ti}^1, u_{ti}^2, v_{ti}^0, v_{ti}^1, u_{tij}^0, u_{tij}^1} \quad g(u^0, v^0, u^1, v^1, \xi, \eta, X, d) \tag{3.2a}$$

$$= \sum_{t\in[T], i\in[n]} (u_{ti}^0 + u_{ti}^1) x_{ti} \tag{3.2b}$$

$$+ \sum_{t\in[T], i\in[n]} \left(\sum_{j\in[n]} d_j x_{tj} \right) u_{ti}^2 \tag{3.2c}$$

$$+ \sum_{t\in[T], i\in[n], j\in[n]} (1 - d_j x_{tj}) u_{tij}^0 \tag{3.2d}$$

$$+ \sum_{t\in[T], i\in[n], j\in[n]\setminus\{i\}} (1 - x_{tj}) u_{tij}^1 \tag{3.2e}$$

$$+ \sum_{i\in[n]} (v_i^0 + v_i^1) + \sum_{t\in[T], i\in[n]} (v_{ti}^0 + v_{ti}^1) \tag{3.2f}$$

$$\text{s.t.} \quad u_{ti}^0 + \sum_{j\in[n]} u_{tij}^0 + v_{ti}^0 - \xi_i \geq 0, \quad \forall t \in [T], i \in [n], \tag{3.2g}$$

$$v_i^0 - \sum_{t\in[T], j\in[n]\setminus\{i\}} u_{tij}^1 + \xi_i \geq 1, \quad \forall i \in [n], \tag{3.2h}$$

$$u_{ti}^1 + u_{ti}^2 + \sum_{j\in[n]\setminus\{i\}} u_{tij}^1 + v_{ti}^1 - \eta_i \geq 0, \quad \forall t \in [T], i \in [n], \tag{3.2i}$$

$$v_i^1 + \eta_i \geq 1, \quad \forall i \in [n], \tag{3.2j}$$

$$v_i^0, v_i^1, \xi_i, \eta_i \geq 0, \forall i \in [n], \tag{3.2k}$$

$$u_{ti}^0, u_{ti}^1, u_{ti}^2, v_{ti}^0, v_{ti}^1 \geq 0, \forall t \in [T], i \in [n], \tag{3.2l}$$

$$u_{tij}^0, u_{tij}^1 \geq 0, \forall t \in [T], i \in [n], j \in [n], \tag{3.2m}$$

Then, we have the following proposition:

Proposition 4. The optimal value of Problem (3.2) is equal to $Q(X, d)$.

Proof. The proof follows from the strong duality of linear programming. □

In order to solve Problem (2.8) using the Benders' decomposition method, we equivalently reformulate it as follows.

$$
\begin{aligned}
\min_{X,b,R,u^0,v^0,u^1,v^1,\xi,\eta} \quad & \sum_{t\in[T]} b_t + n - R && \text{(3.3a)}\\
\text{s.t.} \quad & \sum_{i\in[n]} x_{ti} \le U b_t, \forall t \in [T], && \text{(3.3b)}\\
& \sum_{t\in[T]} x_{ti} \le r, \forall i \in [n], && \text{(3.3c)}\\
& g(u^{0,k}, v^{0,k}, u^{1,k}, v^{1,k}, \xi^k, \eta^k) \ge R, \\
& \quad \forall (u^{0,k}, v^{0,k}, u^{1,k}, v^{1,k}, \xi^k, \eta^k) \in \mathcal{F}, k \in [K], && \text{(3.3d)}\\
& x_{ti} \in \{0,1\}, \forall t \in [T], i \in [n], && \text{(3.3e)}\\
& b_t \in \{0,1\}, \forall t \in [T], && \text{(3.3f)}
\end{aligned}
$$

where $\mathcal{F}$ is the set of all extreme points in Problem (3.2). Problem (3.3) is called the Benders Master Problem where constraints (3.3d) are called optimality cuts. Given the group design matrix X derived from the master problem, we calculate the number of individual tests we need to detect all remaining defective and non-defective items in Problem (2.7). Here, we don't incorporate feasibility cuts because Problem (3.2) is always feasible for any state $d \in \{0,1\}^n$ and test design $X \in \{0,1\}^{T\times n}$. Problem (3.3) will become more difficult to solve when there is a larger number of extreme points. To overcome this difficulty, we use a subset of constraints (3.3d) to solve a relaxed counterpart of Problem (3.3) as follows.

$$
\begin{aligned}
\min_{X,b,R,u^0,v^0,u^1,v^1,\xi,\eta} \quad & \sum_{t\in[T]} b_t + n - R && \text{(3.4a)}\\
\text{s.t.} \quad & \sum_{i\in[n]} x_{ti} \le U b_t, \forall t \in [T], && \text{(3.4b)}\\
& \sum_{t\in[T]} x_{ti} \le r, \forall i \in [n], && \text{(3.4c)}\\
& g(u^{0,k}, v^{0,k}, u^{1,k}, v^{1,k}, \xi^k, \eta^k) \ge R, \\
& \quad \forall (u^{0,k}, v^{0,k}, u^{1,k}, v^{1,k}, \xi^k, \eta^k) \in \mathcal{F}^{'}, k \in [K], && \text{(3.4d)}\\
& x_{ti} \in \{0,1\}, \forall t \in [T], i \in [n], && \text{(3.4e)}\\
& b_t \in \{0,1\}, \forall t \in [T], && \text{(3.4f)}
\end{aligned}
$$

where $\mathcal{F}^{'} \subseteq \mathcal{F}$.

3.3 Algorithm

Based on Problem (3.3), we propose the following Benders' decomposition method for Problem (2.5), where we set both upper bounds for the number of Bender cuts and the termination gap between the optimal values of Problem (2.6) and Problem (3.4). We set this gap to quantify how well Bender cuts (3.4d) can approximate Problem (3.3) within the largest iteration number along the search tree. Although a sufficiently large number of search iterations can always ensure the global convergence of our algorithm, in practice, it isn't computationally efficient because algorithm could already get the optimal solution within much lower time limit. Thus, this stopping criterion can ensure a good balance between the optimality and the computational efficiency.

Algorithm 1 Benders' decomposition method for Problem (2.5)

Initialize: Set stopping tolerance $\epsilon > 0$, iteration counter $l \leftarrow 0$, the maximum iteration number $l_{max} > 0$, and the bounds $\{\mathrm{UB}^{(l)}, \mathrm{LB}^{(l)}\}$; Initialize the solutions $\{X^*, b^*\}$, $\mathcal{F}'^{(l)} \leftarrow \emptyset$, and $\mathcal{O}'^{(l)} \leftarrow \emptyset$;
while $|\mathrm{UB}^{(l)} - \mathrm{LB}^{(l)}| > \epsilon$ and $l \leq l_{max}$ **do**
    Update iteration counter: $l \leftarrow l + 1$;
    Solve Problem (3.4);
    if Problem (3.4) is feasible **then**
        Let $\{X^{(l)}, b^{(l)}, R^{(l)}\}$ be the optimal solutions of Problem (3.4) and $\mathrm{LB}^{(l)}$ the optimal value of Problem (3.4);
    else
        Return solutions $\{X^*, b^*\}$;
    end if
    Letting $X \leftarrow X^{(l)}$, $b \leftarrow b^{(l)}$ and $R \leftarrow R^{(l)}$, we solve Problem (3.2);
    if $\min_k Q(X, d^k) < R$ **then**
        Generate *optimality cuts*:

$$g(u^{0,k,l}, v^{0,k,l}, u^{1,k,l}, v^{1,k,l}, \xi^{k,l}, \eta^{k,l}, X, d^k) \geq R, \forall k \in [K],$$

        where $(u^{0,k,l}, v^{0,k,l}, u^{1,k,l}, v^{1,k,l}, \xi^{k,l}, \eta^{k,l})$ is the optimal solution of Problem (3.2);
    end if
    Let $\mathcal{F}'^{(l)} \leftarrow \mathcal{F}'^{(l-1)} \cup \{(u^{0,k,l}, v^{0,k,l}, u^{1,k,l}, v^{1,k,l}, \xi^{k,l}, \eta^{k,l})\}$;
    if $\mathrm{UB}^{(l)} > \sum_{t\in[T]} b_t^{(l)} - \min_k Q(X^{(l)}, d^k)$ **then**
        Let $\mathrm{UB}^{(l)} \leftarrow \sum_{t\in[T]} b_t^{(l)} - \min_k Q(X^{(l)}, d^k)$, $X^* \leftarrow X^{(l)}$ and $b^* \leftarrow b^{(l)}$;
    end if
end while
Return solutions $\{X^*, b^*\}$;

The Benders' decomposition method converges to the global optimality when outer linearization can well approximate the second stage objective function with Benders cuts. When T and n becomes large, it could take the Benders' decomposition method more time to converge to the global optimality. However, the Benders' decomposition method could run much faster than the binary programming approach when K becomes large because constraints (3.4d) are generated iteratively. In practice, every time a new solution to Problem (3.4) is found, a new branch-and-bound tree is built and enumerated for us to get better solution and update Benders cuts. This process is time consuming because it requires the algorithm to revisit all previously obtained solutions. In order to overcome this computational difficulty, we use branch-and-check method (Thorsteinsson, 2001) based on a single search tree where Bender cuts are generated across the nodes of that search tree. This method is computationally efficient for solving our Problem (2.6) because Problem (3.3) is more difficult to solve than Problem (2.7) after linear programming relaxation. In addition, we only need to add a single optimality cut corresponding to the worst-case scenario of unknown state d. Thus, it can also help improve the computational efficiency of the Benders' decomposition method.

Chapter 4

Numerical Experiment

In this chapter, we compare the performances of the Binary programming (BP) for Problem (2.9) and the Benders' decomposition (BD) method for Problem (3.3) in solving Problem (2.6) under different combinations of parameters K, T and n. More specifically, we need to compare the optimality of these two methods in solving Problem (2.6) and the corresponding computational time. The main purpose of this experiment is to test our theoretical conjecture about the computational efficiency of these two methods.

For each K, we randomly generate ten different instances of following two different discrete uncertainty sets D:

$$D_1 = \left\{ d \in \{0,1\}^n \middle| \sum_{i=1}^{n} d_i = \lfloor \frac{n}{2} \rfloor \right\}$$

and

$$D_2 = \left\{ d \in \{0,1\}^n \middle| \sum_{i=1}^{n} d_i = 3 \right\}.$$

More specifically, we randomly generate the uncertainty sets as follows. i) We set $d_i = 0$ for $i \in [n]$; ii) Two different natural number sequences whose sums are, respectively, equal to $\lfloor \frac{n}{2} \rfloor$ and 3 are sampled without replacement from the integer sequence ranging from 1 to n; iii) We set $d_i = 1$ corresponding to the index sequences generated before. For all the experiments, both methods are coded in Python 3.12 and run on an Intel Core i5 PC with 3 GHz and 16GB RAM. We solve Problems (2.9) and (3.4) using Gurobi 11.0 within the time limit 1800s. For the termination criterion of the Benders' decomposition method, we set $\epsilon = 1$ and $l_{max} = 100$. For constraints (2.5b) and (2.5c), we, respectively, set U and r to be T and n. Here, we set n to be 10 and T to be 5. However, we vary the value of K because we want to evaluate the impact of linear programming relaxation and outer linearization on improving the computational efficiency of the Benders' decomposition method.

Table 4.1: Optimal values obtained by the Binary programming (BP) and the Benders' decomposition (BD) when $D = D_1$, $n = 10$ and $T = 5$.

	BP			BD		
K	Ave	Min	Max	Ave	Min	Max
10	10	10	10	10	10	10
20	10	10	10	10	10	10
30	10	10	10	10	10	10
40	10	10	10	10	10	10
50	10	10	10	10	10	10

Table 4.2: Computational times of the Binary programming (BP) and the Benders' decomposition (BD) when $D = D_1$, $n = 10$ and $T = 5$.

	BP			BD		
K	Ave (s)	Min (s)	Max (s)	Ave (s)	Min (s)	Max (s)
10	26.256	11.550	48.445	15.092	13.974	17.696
20	74.652	37.529	132.765	27.343	26.527	28.094
30	145.716	53.906	354.363	40.076	38.889	42.845
40	169.022	92.795	549.361	54.001	51.177	59.121
50	852.351	516.053	1800.054	62.947	62.386	63.718

Table 4.3: Optimal values obtained by the Binary programming (BP) and the Benders' decomposition (BD) when $D = D_2$, $n = 10$ and $T = 5$.

	BP			BD		
K	Ave	Min	Max	Ave	Min	Max
10	8.6	8	9	10	10	10
20	9.4	9	10	10	10	10
30	10	10	10	10	10	10
40	10	10	10	10	10	10
50	10	10	10	10	10	10

Table 4.4: Computational times of the Binary programming (BP) and the Benders' decomposition (BD) when $D = D_2$, $n = 10$ and $T = 5$.

	BP			BD		
K	Ave (s)	Min (s)	Max (s)	Ave (s)	Min (s)	Max (s)
10	30.111	12.976	47.676	13.179	12.906	13.529
20	329.576	127.496	768.724	26.652	26.191	26.912
30	461.993	307.963	738.292	39.171	38.483	39.765
40	901.024	381.638	1800.036	52.013	50.238	55.459
50	274.661	116.392	558.519	61.663	59.483	64.946

Tables 4.1 and 4.3 show the optimal values obtained by two methods in solving Problem (2.6), where the three columns "Ave", "Min" and "Max" represent the average, minimal and maximal optimal values among ten runs under different uncertainty sets and combinations of n and K. Optimal value is equal to the total number of defective and non-defective items that can be detected by these two solution methods under the specified the stopping criterion. From Tables 4.1 and 4.3, we can see that, given the same time limit, the Benders' decomposition method can obtain the same optimal value as the binary programming when K is larger than 20. Tables 4.2 and 4.4 show the computational times of these two methods in solving Problem (2.6), where the three columns "Ave (s)", "Min (s)" and "Max (s)" represent the average, minimal and maximal computational times among ten runs under different uncertainty sets and combinations of n and K. From Tables 4.2 and 4.4, we can conclude that for all randomly generated uncertainty sets, it takes the Benders' decomposition method less time than the Binary programming to get the same optimal value when K is larger than 10. This finding matches our previous conjecture that the Benders' decomposition method can overwhelm the Binary programming because it can circumvent the computational difficulty caused by the large number of constraints with binary variables especially when K is large.

Chapter 5

Conclusion

In this thesis, we study the joint group design and testing problem by building a two-stage robust optimization model where we develop an optimization based approach to our group testing procedure. Although it can save the cost of individual testing, this model is difficult to solve because of both the robust counterpart of the total number of certified defective and non-defective items and the large number of binary constraints and variables. In order to circumvent this difficulty, we first show the tightness of the linear programming relaxation of the second stage problem. Then, we develop a Benders' decomposition method to solve our model. Finally, we numerically compare our Benders' decomposition method with the benchmark binary programming reformulation of the two-stage robust optimization model. The numerical results show that the Benders' decomposition method is computationally more efficient than the binary programming approach especially when the number of scenarios in our discrete uncertainty set is very large.

In our current work, we only consider the finite discrete uncertainty set in our robust counterpart of group testing problem. However, when we have knowledge of the underlying distribution of test items, we can consider more generalized uncertainty sets like Wasserstein ambiguity sets. Then, it could lead to a more variety of the reformulation of Problem (2.6) for which we also need to develop efficient solution algorithms. In addition, our current numerical experiments are restricted within a small range of parameters, which prevents its application in a large scale test sample region.

In the future, we will try to extend our model and algorithm under other uncertainty sets and perform a more comprehensive numerical experiment accordingly.

Bibliography

Aldridge, M., Johnson, O., & Scarlett, J. (2019). Group testing: An information theory perspective. *Foundations and Trends® in Communications and Information Theory*, *15*(3-4), 196–392. https://doi.org/10.1561/0100000099

Barg, A., & Mazumdar, A. (2015). Group testing schemes from codes and designs. *IEEE Transactions on Information Theory*, *63*, 7131–7141. https://doi.org/10.1109/TIT.2017.2746564

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*(1), 238–252. https://doi.org/10.1007/BF01386316

Chan, C. L., Jaggi, S., Saligrama, V., & Agnihotri, S. (2012). Non-adaptive group testing: Explicit bounds and novel algorithms. *2012 IEEE International Symposium on Information Theory Proceedings*, 1837–1841. https://doi.org/10.1109/ISIT.2012.6283597

Coja-Oghlan, A., Gebhard, O., Hahn-Klimroth, M., & Loick, P. (2021). Optimal group testing. *Combinatorics, Probability and Computing*, *30*(6), 811–848. https://doi.org/10.1017/S096354832100002X

Colbourn, C., & Dinitz, J. (2006). *Handbook of combinatorial designs*. Taylor & Francis. https://doi.org/10.1201/9781420010541

Dantzig, G. B., & Wolfe, P. (1960). A decomposition principle for linear programs. *Operations Research*, *8*(1), 101–111.

Dorfman, R. (1943). The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics*, *14*(4), 436–440. https://doi.org/10.1214/aoms/1177731363

Du, D.-Z., & Hwang, F. K. (1999). *Combinatorial group testing and its applications* (2nd). WORLD SCIENTIFIC. https://doi.org/10.1142/4252

D'yachkov, A. G., Vorobyev, I., Polyanskii, N., & Shchukin, V. (2016). On a hypergraph approach to multistage group testing problems. *2016 IEEE International Symposium on Information Theory (ISIT)*, 1183–1191. https://doi.org/10.1109/ISIT.2016.7541486

Long, D., QI, J., Sun, Y., & Zhang, A. (2022). Distributionally robust group testing with correlation information. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4284685

Malioutov, D., & Malyutov, M. (2012). Boolean compressed sensing: Lp relaxation for group testing. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3305–3308. https://doi.org/10.1109/ICASSP.2012.6288622

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817. https://doi.org/https://doi.org/10.1016/j.ejor.2016.12.005

Scarlett, J. (2019). Noisy adaptive group testing: Bounds and algorithms. *IEEE Transactions on Information Theory*, *65*(6), 3646–3661. https://doi.org/10.1109/TIT.2018.2883604

Thorsteinsson, E. S. (2001). Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In T. Walsh (Ed.), *Principles and practice of constraint programming — cp 2001* (pp. 16–30). Springer Berlin Heidelberg.

Vorobyev, I. (2019). A new algorithm for two-stage group testing. *2019 IEEE International Symposium on Information Theory (ISIT)*, 101–105. https://doi.org/10.1109/ISIT.2019.8849718